



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/527,998	01/27/2006	Reiner Hammerich	09700.0116-00	6380
60668	7590	01/08/2008		
SAP / FINNEGAN, HENDERSON LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			EXAMINER HOANG, SON T	
			ART UNIT 2165	PAPER NUMBER
			MAIL DATE 01/08/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/527,998	Applicant(s) HAMMERICH, REINER	
	Examiner Son T. Hoang	Art Unit 2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 22 October 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-21, and 24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-21 and 24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 21 March 2005 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☒ Some * c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Response to Amendment

1. This communication is in response to the amendment filed on October 22, 2007.

Abstract's objection is withdrawn.

112, 2nd paragraph rejections are withdrawn.

Claims 1-21 and **24** have been amended.

Claims 1-21, and **24** are pending in this Office Action.

Response to Arguments

2. After further search and a thorough examination of the present application, **claims 1-21** and **24** remain rejected.

First, Applicant arguments towards independent **claim 1** regarding the fact that Kesler (Pat. No. US 7,062,502, filed on December 28, 2001) does not teach or suggest a variable in the application program that comprises both the basic data type and the metadata.

In response to the Applicant's arguments, the Examiner respectfully submits in particular. Accordingly, Kesler clearly teaches each of the entity fields within a database table is assigned a single data entry control and each entity field has its own metadata to enforce the type of allowed inputs in each field (*text*, *numeric*, *alphanumeric* etc. See *further definition of metadata in Table 1 of Column 4*). The metadata relating to each entity field defines the behavior of its corresponding data entry control ([Column 10, Lines 30-35]). Kesler further teaches in Figure 32 the process of selecting appropriate control for each data input type of the entity field based on the corresponding entity

field's metadata. For example, if the data type is numeric, the '*NumericFormatBox*' control will be used. If the data type is text and text length is greater than 255 characters, then the '*NotesControl*' control will be used. It's clearly shown that each data control entry is a variable that comprises both data type and the metadata ([Column 16, Lines 46-56]).

Second, Applicant argument towards dependent **claim 9** regarding the fact that Logan et al. (Pub. No. US 2003/0093790, filed on June 8, 2002, hereinafter Logan) does not teach storing the application program using the metadata and the metadata in the private database.

In response to the Applicant's arguments, the Examiner respectfully submits in particular. Accordingly, Logan teaches the playback control 211 of the Personal Video Recorder (PVR) controls the playback of stored video programming seen at 217, stored electronic program guide (EPG) data seen at 218, application data such as standard templates stored at 220, metadata describing programs and program segments stored at 221, and other system control data stored at 222 ([0301]-[0304]). This clearly shows that the metadata and the application program that uses metadata are stored locally in a PVR device.

Interpretation of Claims-Broadest Reasonable Interpretation

During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.' Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the

Examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

Reference is made to MPEP 2144.01 - Implicit Disclosure

"[I]n considering the disclosure of a reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." In re Preda, 401 F.2d 825, 826, 159 USPQ 342, 344 (CCPA 1968)

Subsequent to an analysis of the claims it was revealed that a number of limitations recited in the claims belong in the prior art and thus encompassed and/or implicitly disclosed in the reference (s) applied and cited. It is logical for the Examiner to focus on the limitations that are "crux of the invention" and not involve a lot of energy and time for the things that are not central to the invention, but peripheral. The Examiner is aware of the duties to address each and every element of claims, however, it is also important that a person prosecuting a patent application before the Office or an stakeholders of patent granting process make effort to understand the level of one of ordinary skill in the (data processing) art or the level one of skilled in the (data processing) art, as encompassed by the applied and cited references. The administrative convenience derived from such cooperation between the attorneys and Examiners benefits the Office as well the patentee.

In view of the above, the Examiner contends that all limitations as recited in the claims have been addressed in this Action.

For the above reasons, the Examiner believed that rejection of the last Office action was proper.

Hence, Applicant's arguments do not distinguish over the claimed invention over the prior art of record.

In light of the foregoing arguments, the 35 U.S.C. 102 and 103 rejections are hereby sustained.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate Paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this Section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the Applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the Applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. **Claims 1-3, 5-8, 11-14, 16-19, 24**, are rejected under 35 U.S.C. 102(e) as being anticipated by Kesler (Pat. No. US 7,062,502, filed on December 28, 2001).

Regarding **claim 1**, Kesler clearly shows and discloses a computer-implemented method for dynamic data type enrichment (*Abstract*) comprising the steps:

using at least one basic data type in a predefined application program (*The MaskedTextBox is designed to provide an input mask for text fields that store formatted data such as social security numbers*, [Column 35, Line 59 → Column 36, Line 5], see also *CheckBox, ComboBoxControl*,

DateTimeControl, NumericFormatBox in [Column 34, Line 56 → Column 36, Line 23] for further illustrations); and

adding metadata to the at least one basic data type at runtime when the application program is executed to create a variable in the application program (*The Mask Type attribute defines the type of mask to be used. A few predefined masks are available such as PHONE_MASK and SSN_MASK. The GENERAL_MASK Mask Type is used in combination with the Format entity field attribute to define custom input masks using a set of special characters. For example, setting the Format attribute to "A###" generates an input mask that ensures the first character must be alphanumeric and the last three characters must be digits. The SSN field in the Employee entity uses the SSN_MASK Mask Type as illustrated in Fig. 51, [Column 35, Line 59 → Column 36, Line 5]) that comprises both the basic data type and the metadata (Figure 32 shows the process of selecting appropriate control for each data input type of the entity field based on the corresponding entity field's metadata. For example, if the data type is numeric, the 'NumericFormatBox' control will be used. If the data type is text and text length is greater than 255 characters, then the 'NotesControl' control will be used. It's clearly shown that each data control entry is a variable that comprises both data type and the metadata ([Column 16, Lines 46-56]).*

Regarding **claim 2**, Kesler further discloses a method, wherein the application program uses an application programming interface for accessing the metadata before adding the metadata to the basic data type (*Figure 58 shows Configuration Repository file where all the metadata are configured*).

Regarding **claim 3**, Kesler further discloses a method, wherein the application program calls through the application programming interface at least one metadata service that relates to the basic data type (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the users would know when they entered the wrong format as illustrated in Figure 51*).

Regarding **claim 5**, Kesler further discloses a method, wherein the basic data type is defined in a programming language used by the application program (*Social Security Number is a sequence of integers used by any known programming language*).

Regarding **claim 6**, Kesler further discloses a method, wherein the metadata is associated with a specific data type defined in a metadata store (*Information gleaned from the schema is used to populate a configuration repository with metadata. This metadata is then utilized by the user interface to generate data entry forms, [Column 8, Lines 1-12]*).

Regarding **claim 7**, Kesler further discloses a method, wherein the application program provides a mapping between the specific data type and the basic data type (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the users would know when they entered the wrong format as illustrated in Figure 51, see also [Column 38, Line 66 → Column 39, Line 4] for further illustrations*).

Regarding **claim 8**, Kesler further discloses a method, wherein the application program uses a variable to map the specific data type to the basic data type (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the users would know when they entered the wrong format as illustrated in Figure 51, see also [Column 38, Line 66 → Column 39, Line 4] for further illustrations*).

Regarding **claim 11**, Kesler clearly shows and discloses a computer program product comprising instructions embodied on a memory of a computer system cause at least one processor of the computer system to execute the steps of **claim 1** (*Computer software for, apparatus for, and a method of*

automatically extracting schema information and generating corresponding schema and user interface metadata; storing the metadata in a repository and automatically developing from the metadata a user interface appropriate to the relational database, [Column 2, Line 49 → Column 4 Line 7]) .

Regarding **claim 12**, Kesler clearly shows and discloses a computer system ([Column 2, Line 49 → Column 5, Line 7]) comprising:

a memory ([Column 2, Line 49 → Column 5, Line 7]) storing an application program that uses a basic data type (*The MaskedTextBox is designed to provide an input mask for text fields that store formatted data such as social security numbers, [Column 35, Line 59 → Column 36, Line 5]); and*

a processor ([Column 2, Line 49 → Column 5, Line 7]) executing instructions to add metadata to the basic data type when executing the application program to create a variable in the application program (*The Mask Type attribute defines the type of mask to be used. A few predefined masks are available such as PHONE_MASK and SSN_MASK. The GENERAL_MASK Mask Type is used in combination with the Format entity field attribute to define custom input masks using a set of special characters. For example, setting the Format attribute to "A###" generates an input mask that ensures the first character must be alphanumeric and the last three characters must be digits. The SSN field in the Employee*

entity uses the SSN_MASK Mask Type as illustrated in Fig. 51, [Column 35, Line 59 → Column 36, Line 5]) that comprises both the basic data type and the metadata (Figure 32 shows the process of selecting appropriate control for each data input type of the entity field based on the corresponding entity field's metadata. For example, if the data type is numeric, the 'NumericFormatBox' control will be used. If the data type is text and text length is greater than 255 characters, then the 'NotesControl' control will be used. It's clearly shown that each data control entry is a variable that comprises both data type and the metadata ([Column 16, Lines 46-56]).

Regarding **claim 13**, Kesler further discloses a computer system further comprising an application programming interface to access the metadata from the application program (*Figure 58 shows Configuration Repository file where all the metadata are configured*).

Regarding **claim 14**, Kesler further discloses a computer system, wherein the application programming interface provides at least one metadata service that relates to the basic data type used by the application program (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the*

users would know when they entered the wrong format as illustrated in Figure 51).

Regarding **claim 16**, Kesler further discloses a computer system, wherein the basic data type is defined in a programming language used by the application program (*Social Security Number is a sequence of integers used by any known programming language*).

Regarding **claim 17**, Kesler further discloses a computer system, wherein the metadata are associated with a specific data type defined in a metadata store (*Information gleaned from the schema is used to populate a configuration repository with metadata. This metadata is then utilized by the user interface to generate data entry forms, [Column 8, Lines 1-12]*).

Regarding **claim 18**, Kesler further discloses a computer system, wherein the application program provides a mapping between the specific data type and the basic data type (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the users would know when they entered the wrong format as illustrated in Figure 51, see also [Column 38, Line 66 → Column 39, Line 4] for further illustrations*).

Regarding **claim 19**, Kesler further discloses a computer system, wherein the application program uses a variable to map the specific data type to the basic data type (*The mapping process as described in explanations of claim 1. Accordingly, when Social Security Number field is identified, the application program will go back to the repository to relate the appropriate metadata for that particular data type. From the metadata, a particular format of Social Security Number will be applied so the users would know when they entered the wrong format as illustrated in Figure 51, see also [Column 38, Line 66 → Column 39, Line 4] for further illustrations).*

Regarding **claim 24**, Kesler clearly shows and discloses a method for changing metadata (*Abstract*) comprising the steps:

executing an application program that uses at least one metadata service to access the metadata in a metadata store (*Data entry and navigation functionality in the UI are provided through "pop-up" menus. These menus are generated dynamically from metadata stored in the configuration repository. The metadata associated with menus is "pre-defined" in the sense that it is not derived from the database schema but instead built in to the UI architecture, [Column 12, Lines 20-36]*);

changing the metadata in the metadata store at runtime of the application program (*When add or remove a field, the metadata relating to an entity's collection of fields is simply modified within the Utility Tool to*

reflect the change. The data entry form will automatically adjust to changes in metadata without the necessity of writing any computer code, [Column 17, Lines 19-25]); and

using the at least one metadata service in the application program for using the changed metadata without restarting the application program (Scripting is stored as metadata in the configuration repository, and does not require recompilation and redeployment of the UI software when changes are made, [Column 33, Lines 28-43]) to create a variable in the application program (The Mask Type attribute defines the type of mask to be used. A few predefined masks are available such as PHONE_MASK and SSN_MASK. The GENERAL_MASK Mask Type is used in combination with the Format entity field attribute to define custom input masks using a set of special characters. For example, setting the Format attribute to "A###" generates an input mask that ensures the first character must be alphanumeric and the last three characters must be digits. The SSN field in the Employee entity uses the SSN_MASK Mask Type as illustrated in Fig. 51, [Column 35, Line 59 → Column 36, Line 5]) that comprises both the basic data type and the metadata (Figure 32 shows the process of selecting appropriate control for each data input type of the entity field based on the corresponding entity field's metadata. For example, if the data type is numeric, the 'NumericFormatBox' control will be used. If the data type is text and text length is greater than 255

characters, then the 'NotesControl' control will be used. It's clearly shown that each data control entry is a variable that comprises both data type and the metadata ([Column 16, Lines 46-56]).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 4, and 15** are rejected under 35 U.S.C. 103(a) as being unpatentable over Kesler (Pat. No. US 7,062,502, filed on December 28, 2001) in view of Koseki et al. (Pat. No. US 6,732,124, filed on February 9, 2000, hereinafter Koseki).

Regarding **claim 4, and 15** Kesler does not explicitly disclose a metadata service copies the metadata to a metadata cache.

Koseki discloses a metadata cache is provided as part of the computer's main memory to hold a copy of metadata objects from the metadata volume. A metadata loading unit reads out a specific metadata object from the metadata volume to the metadata cache when a transaction demands it ([Column 9, Lines 48-58]).

It would be obvious to a person skilled in the art at the time of the invention to incorporate the teachings of Koseki with the teachings of Kesler for the purpose of preventing unwanted changes in the original metadata by

modifying the copy of metadata in the metadata cache instead of directly manipulating the original in the metadata volume ([Column 9, Lines 55-58] of Koseki).

7. **Claims 9-10, 20-21** are rejected under 35 U.S.C. 103(a) as being unpatentable over Kesler (Pat. No. US 7,062,502, filed on December 28, 2001) in view of Logan et al. (Pub. No. US 2003/0093790, filed on June 8, 2002, hereinafter Logan).

Regarding **claims 9-10, 20-21** Kesler does not explicitly disclose the metadata is stored along with the application program in private instance of the metadata. Also, Kesler does not explicitly disclose storing metadata in public instance of the metadata.

Logan discloses that metadata contributed by other users and stored in a public database as well as private database. The metadata stored may be created, edited and deleted using a Web server or other server operates to contribute to the metadata ([0308]). Lohan further discloses the playback control 211 of the Personal Video Recorder (PVR) controls the playback of stored video programming seen at 217, stored electronic program guide (EPG) data seen at 218, application data such as standard templates stored at 220, metadata describing programs and program segments stored at 221, and other system control data stored at 222 ([0301]-[0304])

It would be obvious to a person skilled in the art at the time of the invention to incorporate the teachings of Logan with the teachings of Kesler for the purpose of enhancing user's enjoyment of available broadcast programming

content by utilizing metadata created either at a central location for shared used by connected users, or at each individual user's location ([Abstract] of Logan).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Communication

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Son T. Hoang whose telephone number is (571) 270-1752. The Examiner can normally be reached on Monday - Friday.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Christian Chace can be reached on (571) 272-4190. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.


CHRISTIAN CHACE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100



/S.H./
Son T. Hoang
Patent Examiner
January 3, 2008